Computer Science Department

TECHNICAL REPORT

Implementation of the Schnabel & Eskow Modified Cholesky Factorization in the Context of a Truncated-Newton Optimization Method

Tamar Schlick

Technical Report 525

October 1990

NEW YORK UNIVERSITY

Department of Computer Science
Courant Institute of Mathematical Sciences
251 MERCER STREET, NEW YORK, N.Y. 10012

Implementation of the Schnabel & Eskow Modified Cholesky Factorization in the Context of a Truncated-Newton Optimization Method

Tamar Schlick

Technical Report 525

October 1990

Implementation of the Schnabel & Eskow Modified Cholesky Factorization in the Context of a Truncated-Newton Optimization Method

Tamar Schlick
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012

ABSTRACT

We report our experience with an application of the new modified Cholesky Factorization of Schnabel & Eskow in the context of nonlinear optimization. We have implemented a modified version of the new factorization for sparse symmetric linear systems, without pivoting, and have incorporated it into the framework of a large-scale truncated Newton minimization package. In our truncated Newton algorithm, we use the preconditioned linear conjugate gradient method to solve iteratively and approximately for the Newton search direction. The MCF is used here to guarantee that the effective preconditioner is positive definite. Details of the implementation are provided, and performance comparisons with the Gill, Murray and Wright modified Cholesky factorization for sample problems are reported. Our preliminary results demonstrate that the two modified Cholesky factorizations perform quite differently in practice in our truncated Newton context. A clear difference in timing of the modification (i.e., first nonzero increment), in addition to differences in numerical values, suggests that the Schnabel & Eskow strategy may work better for highly indefinite systems while the Gill, Murray and Wright strategy may be more effective for nearly positive definite systems. As a consequence, the new factorization may be especially useful for minimization of highly nonlinear functions by Newton methods.

This work is made possible through generous support from the National Science Foundation, New York State Science and Technology Foundation, the San Diego Supercomputer Center, and the Academic Computing Facility at New York University. T. Schlick is recipient of the Marie Curie American Fellowship from the American Association of University Women Educational Foundation.

Implementation of the Schnabel & Eskow Modified Cholesky Factorization in the Context of a Truncated-Newton Optimization Method

I. INTRODUCTION

The purpose of this communication is to report our experience with an application of the new modified Cholesky Factorization (MCF) of Schnabel & Eskow [1,2] in the context of our nonlinear optimization package. We have implemented a modified version of the new factorization for large sparse symmetric linear systems, without pivoting, and have incorporated it into the framework of a large-scale truncated Newton minimization algorithm [3]. In our truncated Newton algorithm, we approximate the solution to the Newton equations at every step by using an iterative, linear Conjugate Gradient method (CG). The CG loop is "truncated" when either: 1) a suitably chosen truncation criterion is satisfied, or 2) indefiniteness is detected, in which case a guaranteed direction of descent is still produced. The CG loop is accelerated through preconditioning. To guarantee that the effective preconditioner — often chosen from the physics of the problem — is positive definite, we use the modified Cholesky factorization. In this report, details of the implementation will be provided, and performance comparisons with the Gill, Murray and Wright MCF [4] for sample computational chemistry problems and the trigonometric function will be reported. Results demonstrate that the two modified Cholesky factorizations perform quite differently in practice and that their relative performance is sensitive to problem characteristics related to indefiniteness. A clear difference in timing of modifications (i.e., first nonzero increment) suggests that the Schnabel & Eskow strategy may work better for large, highly indefinite systems — especially with clustered negative eigenvalues — while the Gill, Murray & Wright strategy may be more effective for nearly positive definite systems. Thus, the Schnabel & Eskow factorization may be particularly important for minimization of highly nonlinear functions by Newton methods.

II. THE MODIFIED CHOLESKY FACTORIZATION

The recently-reported MCF of Schnabel & Eskow (henceforth denoted as S&E) presents an alternative to the Gill, Murray, and Wright MCF (GMW). The GMW factorization has been used extensively to solve linear systems $M\mathbf{z} = \mathbf{r}$ for coefficient matrices M that are symmetric but not necessarily positive definite. Such factorizations are particularly appropriate in the context of nonlinear optimization, and this topic will be addressed in the next section. Effectively, a diagonal matrix E of nonnegative elements is added to M, and the sum $M + E$ is factored as $M + E = LDL^T$ where D is diagonal and L is unit lower-triangular. (M, E, L, and D are all $n \times n$ matrices).

Two important questions in these modified factorizations are the following: 1) how do we perform the MCF in a numerically-stable manner without knowing the complete eigenvalue distribution of M a priori? and 2) how do we choose the appropriate numerical values for the entries of E? The first issue is important because the Cholesky factors may be ill-conditioned and may not exist for an indefinite system; thus, the standard process must be amended appropriately. Furthermore, the algorithm should be formulated so as to keep the computational cost as close as possible to that of the standard Cholesky factorization. In regard to the second question of constructing E, we must generate a clever recipe for choosing the increments e_j, $j = 1, \ldots, n$, so as to balance: 1) on one extreme, selecting large modifications that guarantee positive definiteness but perturb the original matrix excessively; with 2) on the other extreme, choosing small, just sufficient, increments that may lead later to very large increments.

The new S&E is attractive because it possesses a lower a priori upper bound on the size of E — measured as $\|E\|_\infty$ — than the GMW factorization [1,2]. Its cost is comparable to that of GMW, involving an additional multiple of n^2 operations to the standard Cholesky factorization. Furthermore, numerical experiments with S&E have suggested that the resulting $\|E\|_\infty$ may indeed be smaller than GMW in practice [1].

Full details of the two factorizations are provided in the original papers. Below we only summarize the two different procedures for choosing the elements of E. While pivoting strategies are also different for the two factorizations, they will not be addressed here since we are interested in a no-pivoting version. It is worth mentioning, however, that while pivoting does not affect the theoretical upper bounds on $\|E\|_\infty$, it may produce better results in practice.

Consider step j of a row-wise, no-pivoting, modified Cholesky factorization (see Figure 1). We adopt the convention that computed elements of L and D are overwritten in the storage arrays of the original M and that the diagonal array of D has been initialized at step 0 as $d_j \leftarrow m_{jj}$, $j = 1, \ldots, n$. As step j begins, the upper $(j-1)$ by $(j-1)$ submatrix contains the final values for the elements of L and D. Row j up to the diagonal contains the computed "auxiliary" quantities $c_{js} = \ell_{js}\, d_s$, $s = 1, \ldots, j-1$. Similarly, rows $j+1$ through n up to column j contain their auxiliary values c. The candidate for d_j has been computed at the end of step $j-1$. The current step j of the MCF consists of the following substeps:

(a) Update row j:

$$\ell_{js} = c_{js} \,/\, d_s \;, \quad s = 1, \ldots, j-1 \;. \tag{1}$$

(b) Compute the auxiliary quantities c in column j below the diagonal, and set θ to some functional value (to be discussed below) of these quantities:

$$c_{sj} = m_{sj} - \sum_{k=1}^{j-1} \ell_{jk}\, c_{sk} \;, \quad s = j+1, \ldots, n \;, \tag{2}$$

$$\theta = f(\mathbf{c}_j) \;, \quad \mathbf{c}_j = (c_{j+1,j}, c_{j+2,j}, \ldots c_{n,j})^T \;. \tag{3}$$

(c) Modify d_j if necessary by adding a quantity e_j specified by a given recipe (further details are provided later):

$$d_j \leftarrow d_j + e_j \;. \tag{4}$$

(d) Update the prospective diagonal elements:

$$d_s \leftarrow d_s - c_{sj}^2 / d_j \ , \quad s = j+1, \ldots, n \ . \tag{5}$$

The details of substeps (b)-(d) above will now be discussed in turn for the GMW and S&E factorizations.

(A) The GMW Factorization

In GMW, the values for e_j are chosen so as to minimize an *a priori* upper bound on $\|E\|_\infty$ subject to the condition that positive definite matrices will not be perturbed (i.e., M positive definite $\Rightarrow E = 0$).

First, the method requires as input the values of two small numbers: ϵ_m — machine precision, and τ — the lowest acceptable value for $d_j + e_j$, typically of order of square or cubic root of ϵ_m. Second, the following key parameters must be computed before the factorization begins:

$$\gamma = \max_j \{ |m_{jj}| \} \tag{6}$$

$$\xi = \max_{j>s} \{ |m_{js}| \} \tag{7}$$

and the bound

$$\beta = \max \left\{ \gamma \ , \ \frac{\xi}{\max\{1, \sqrt{n^2 - 1}\}} \ , \epsilon_m \right\} . \tag{8}$$

In substep (b) of the MCF as sketched above, the GMW function is: $f_{GMW}(\mathbf{c}) = \|\mathbf{c}\|_\infty$. Thus, θ contains the largest magnitude among the c_{sj}, $s = j+1, \ldots, n$. In substep (c) of the MCF, the GMW recipe for incrementing d_j defines e_j implicitly through the formula:

$$d_j \leftarrow max\{|d_j| \ , \ \tau \ , \ \theta^2/\beta\} \ . \tag{9}$$

The following points are worth noting. First, the matrix M must be assembled a priori for GMW so that γ and ξ can be computed. Thus, some modifications may be required

in some large-scale applications if, typically, the entries of M are assembled row-by-row as the factorization proceeds. Second, even when the candidate for d_j is greater than τ, the value may be modified if it is less than θ^2/β. This situation can occur when M is not sufficiently positive definite. If the relation $\tau < \theta^2/\beta < |d_j|$ holds, then $e_j = 2|d_j|$. Our experience has shown that for indefinite systems this is often the resulting increment. Third, the restriction that $d_j \geq \theta^2/\beta$ implies that

$$d_s \|\ell_{sj}^2\|_\infty \leq \beta \ . \tag{10}$$

Hence the name "bound" for β.

(B) The S&E Factorization

The S&E recipe for adding the e_j's is not applied in a straightforward manner — at every step j — as in GMW. Instead, a two-phase strategy for the entire factorization is used. Phase 1 denotes the standard Cholesky factorization in which all increments are zero. Phase 1 ends when the diagonal test — see below — indicates that some prospective diagonal in the remaining submatrix will become unreasonably small. At this point, the algorithm switches to Phase 2, and the S&E recipe for the increments is applied from this step forward.

Two small parameters must be prescribed by the user for the S&E factorization: τ_1 and τ_2, typically around the same magnitude as the cubic root of machine precision. Now recall that the candidate for d_j, as well as for the updated diagonal elements $\{d_{j+1}, \ldots, d_n\}$ in the remaining $(n-j+1)$ by $(n-j+1)$ submatrix are computed at step $j-1$, substep (d), of the MCF (eq. (5)). In S&E, while we are still in Phase 1, the phase test sets d_{min} at step j to

$$d_{min} \leftarrow \min_{s \geq j} \{d_s\} \tag{11}$$

and tests whether

$$d_{min} < \tau_1 \gamma \ . \tag{12}$$

The parameter γ is the same for GMW, defined by eq. (6). If eq. (12) holds, the S&E algorithm switches to Phase 2, and the e_j recipe is applied.

The function f in substep (b) of the MCF (eq. (3)) is the 1-norm for S&E instead of the ∞-norm as in GMW: $f_{S\&E}(\mathbf{c}) = \|\mathbf{c}\|_1 = \sum_{s=j+1}^{n} |c_{sj}|$. Thus, $\theta = f_{S\&E}(\mathbf{c})$ contains the Gerschgorin radius for row j at the current step of the factorization. The recipe for e_j is then given by:

$$e_j = \max\{0\ ,\ -d_j + \max\{\theta\ ,\ \tau_2\gamma\}\ ,\ e_{j-1}\}\ , \tag{13}$$

where e_{j-1} is the increment at step $j-1$.

For the last two steps (which determine e_n and e_{n-1}), Schnabel & Eskow modify their recipe above by using eigenvalues of the final 2×2 submatrix directly. If $\lambda_{\ell o}$ and λ_{hi} denote the eigenvalues of the remaining submatrix $\begin{bmatrix} d_{n-1} & c_{n,n-1} \\ c_{n,n-1} & d_n \end{bmatrix}$ at the end of step $n-2$, then

$$e_{n-1} = e_n = \max\left\{0\ ,\ e_{n-2}\ ,\ -\lambda_{\ell_0} + \tau_2\ *\ \max\{\frac{1}{1-\tau_2}(\lambda_{hi} - \lambda_{\ell o})\ ,\ \gamma\}\right\}\ . \tag{14}$$

The reasoning behind this Gerschgorin strategy in S&E is that $\|E\|_\infty$ can be shown to be bounded above by the magnitude of the most negative lower Gerschgorin bound of the submatrix remaining at the step where Phase 2 is entered. In combination with an upper bound for Phase 1, this result produces a lower upper bound on $\|E\|_\infty$ for S&E than for GMW.

III. THE MCF IN THE CONTEXT OF OPTIMIZATION

Modified Cholesky factorizations are particularly useful in the context of nonlinear optimization. In iterative Newton-type methods for minimizing $F(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, a sequence of vectors $\{\mathbf{x}_0, \mathbf{x}_1, \ldots\}$ is generated by the rule $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{p}_k$. The vector $\mathbf{p}_k$ is a search direction, and the scalar $\lambda_k > 0$ is a step length that ensures sufficient function decrease. The vector $\mathbf{p}_k$ can be obtained by solving the Newton equations for $\mathbf{p}$,

$$H_k \mathbf{p} = -\mathbf{g}_k \ , \tag{15}$$

where H and $\mathbf{g}$ are the Hessian and gradient vector, respectively, of E at $\mathbf{x}_k$.

In the "modified Newton" framework, developed to work in practice when a local quadratic approximation to F may be poor, a positive definite approximation to H_k may replace H_k in eq. (15). Furthermore, in the truncated Newton variation, the Newton (or modified Newton) equations may only be solved approximately [5,6]. The rationale behind this approximation is that it is necessary to obtain an accurate Newton search direction only near a minimum; far away from such regions, any descent direction will suffice, so the cost for exact solution is unwarranted. Indeed, it has been shown in theory and in practice [5,3,7] that overall convergence is not sacrificed if the truncation criterion is chosen to ensure that, as a critical point of F is approached, the solution for $\mathbf{p}_k$ becomes more accurate. For example, the approximation (to the Newton search direction at step k) can be controlled by a parameter η_k, set to

$$\eta_k = \min\{\delta/k \ , \ \|\mathbf{g}_k\|\} \tag{16}$$

where $\delta \leq 1$, and then the residual is computed to satisfy

$$\|H_k \mathbf{p}_k + \mathbf{g}_k\| \leq \eta_k \|\mathbf{g}_k\| \ . \tag{17}$$

From the discussion above, one obvious utility for the MCF appears: modify H_k when it is not positive definite as $H_k \leftarrow H_k + E_k$ in eqs. (15) or (17). The less obvious

utility for the MCF in our modified Newton framework arises in the truncated Newton implementation.

Truncated Newton methods lead to a nested iteration structure: outer loop for $\mathbf{x}_k$, inner loop for $\mathbf{p}_k$. An iterative, rather than direct, procedure must be used in the inner loop to allow truncation of the solution process for $\mathbf{p}_k$. One attractive method for large-scale problems is the linear preconditioned Conjugate Gradient method (PCG) [8], which must be used with adaptations to indefinite systems [3,5]. Preconditioning is important for accelerating convergence, but it adds yet another linear system, which must be solved at every *inner* iteration: $M\mathbf{z} = \mathbf{r}$, where M is the preconditioner of H, and $\mathbf{z}$ and $\mathbf{r}$ are vectors. Typically, the benefit in performance from preconditioning outweighs the additional complexity involved. Now, if M is not guaranteed to be positive definite, the MCF can be used to modify M as $M \leftarrow M+E$ so that the "effective preconditioner" is positive definite. This context is appropriate in many large-scale applications that arise in computational chemistry, mathematical biology, or meteorology, where a sparse preconditioner can often be extracted from the "physics" of the problem. TNPACK implements the MCF in this way, and further details of the algorithm follow in the next section.

Whether the MCF is used to modify the Hessian or the preconditioner in the modified Newton framework, the following questions arise regarding implementation and testing of the GMW and S&E factorizations in this minimization context:

(1) Can both factorizations be implemented effectively in an optimization code in analogous programming style — so that both factorizations can be exercised?

(2) In particular, can the two factorizations be implemented efficiently in large-scale minimization codes where problem structure and size mandate special storage and algorithmic considerations?

(3) On average over the number of iterations, will $\|E\|_\infty$ be lower for S&E than for GMW?

(4) Whatever the outcome from question (3) above, how will the size of $\|E\|_\infty$ affect

the overall progress toward a local minimum?

In the next section, we summarize some implementation details, and in the following section we describe several numerical experiments.

IV. IMPLEMENTATION OF THE MCF IN TNPACK

Our recently-prepared package TNPACK [3] uses a large scale truncated-Newton algorithm for unconstrained nonlinear optimization. As described above, truncation is accomplished by using PCG to solve approximately the Newton equations so that eqs. (16) and (17) are satisfied. Since PCG was originally developed for positive definite systems, adaptations for the indefinite case must be made in the present context. This involves halting a PCG iteration when a direction of negative curvature ($\mathbf{d}$) is detected: $\mathbf{d}^T H_k \mathbf{d} < \zeta \mathbf{d}^T \mathbf{d}$ where ζ is a tolerance parameter. In this event, we exit the inner PCG loop with a search direction that is always a direction of descent [3,5]. In brief, if this occurs at the first PCG iteration, $\mathbf{p}_k$ is set to $-M^{-1}\mathbf{g}_k$ ($-\mathbf{g}_k$ is another possibility); otherwise, $\mathbf{p}_k$ is set to the current $\mathbf{p}$ in the inner iteration. (Other choices after the first iteration, such as the current $\mathbf{d}$ or the steepest descent direction, $-\mathbf{g}_k$, can be selected by the user).

Next, we have developed a procedure for implementing PCG for large-scale problems where the preconditioner M is large and sparse. Recall that CG methods require Hessian/vector multiplications ($H\mathbf{d}$) at every step, and that PCG algorithms require, in addition, solution of a linear system $M\mathbf{z} = \mathbf{r}$ at every step. Thus M remains constant throughout the inner PCG loop of Newton step k, but several solutions are required for different right-hand-side vectors $\mathbf{r}$. In TNPACK, we solve these linear sytems by a *sparse* MCF.

Our sparse factorization is based on the Yale Sparse Matrix Package, YSMP [9-11]. This package was developed to solve large, sparse systems efficiently by Gaussian elimination. For positive definite systems, efficiency is accomplished by 1) using compact row-by-row storage schemes for M and its Cholesky factors, 2) reordering the rows and

columns of M in positions which were zeros for M, 3) performing arithmetic on the nonzero elements only, and 4) employing a modular solution process. The modularity is accomplished through 3 separate stages — symbolic factorization, numerical factorization, and numerical solution — and allows efficiency in solving several related systems (e.g., same M, same sparsity structure). We have modified two YSMP routines for TNPACK to handle matrices that are symmetric but not necessarily positive definite by incorporating the MCF. Thus, although our target problems are large and their associated Hessians may be dense, the preconditioners are typically sparse and the YSMP is efficient with its compact storage and computational schemes.

(A) No-Pivoting Strategies

In TNPACK, reordering of the variables is provided as an option. Reordering is used to minimize fill-in of the Cholesky factors in M and is performed only once, before the factorization of M begins. In many large-scale optimization problems, a sparse Hessian or a sparse approximation to the Hessian is involved. Often, sparse preconditioners can be easily formulated from the separability structure of the Hessian in question. For example, in image processing applications, neighboring pixels can be assembled. In computational chemistry, energy components from the "local" interactions — bond-length and bond-angle potentials — can be included [12,13]. (The remaining terms are electrostatic and Van der Waals and are computed over *all* atom pairs.) Moreover, in these physical applications, the pattern of M often remains the same throughout the entire optimization process. Thus, reordering of the variables can be performed only once, at the beginning of the minimization algorithm. Every subsequent Newton step will then compute different numerical values for M, and the MCF without pivoting will be used to factor M (once) and then solve for several different right-hand-side vectors in each inner PCG iteration.

(B) The Phase-Test Strategy

When no-pivoting is used in MCF, two sets of different input parameters, values of θ, and recipes for setting the e_j's for the two factorizations can easily be programmed in FORTRAN IF-THEN-ELSE segments (so that both factorizations can be exercised). Less straightforward is implementing the phase test of S&E. If an existing MCF algorithm is written as in Section II — with prospective diagonals updated at every step of the elimination — the phase test is easy to implement. If, on the other hand, the code does not update the diagonals at every step and instead computes each candidate d_j at step j, substep (a) of the code must be modified as:

$$d_j \leftarrow m_{jj} - \sum_{s=1}^{j-1} \ell_{js}^2 \, d_s \; . \tag{18}$$

For large linear systems, the requirement to update the diagonal elements at every step means, of course, that the entire diagonal array be available before the MCF begins. Thus, codes from applications, such as multifrontal methods, that normally assemble matrix entries row-by-row as the elimination proceeds, must be modified to assemble all diagonals at the start. (Off-diagonals may still be computed progressively). The diagonal array is also required to compute γ (eq. (6)). Note that GMW requires, in addition, that the off-diagonal array be available at the start of the factorization in order to compute ξ (eq. (7)).

(C) The Recipe Strategy

A remaining issue to be addressed in implementing S&E in the context of large-scale nonlinear optimization is the Gerschgorin-based formula for setting the e_j's. Schnabel & Eskow propose two modifications which do not change the theoretical bounds on $\|E\|_\infty$ but may, in fact, lead to better results in practice. The first modification is eq. (14) — using actual eigenvalues of the remaining 2×2 submatrix to determine e_n and e_{n-1}. The second modification is including e_{j-1} in eq. (13) for e_j. The reasoning behind this strategy is that

requiring $e_j \geq e_{j-1}$ will not increase $\|E\|_\infty$ but may, in fact, lead to smaller increments at later Cholesky steps [1,2].

Our experience suggests (see next section) that including e_{j-1} in the formula for e_j does not always improve the overall performance in optimization. While $\|E\|_\infty$ may indeed be larger if the condition $e_j \geq e_{j-1}$ is not imposed, this outcome is balanced by the fact that *fewer* overall modifications to M are involved. Moreover, for very large sparse systems, there is an additional argument for omitting this condition: if $\ell_{j+1,j}$ or $c_{j+1,j}$ is zero, a larger d_j will *not* lead to a more negative d_{j+1} and hence to a larger e_{j+1} — see eq. (5).

The second modification of the recipe at the last two steps may have a small improvement, if any, on $\|E\|_\infty$ for large-scale systems. Furthermore, this special treatment of the last two variables may destroy some inherent symmetry of the problem. For example, if an objective function suggests the relation $x_j = x_{j+2}$, $j = 1, 2, \ldots$, it seems reasonable to allow an analogous symmetry in the components of the PCG vectors (which, in turn, will lead to symmetry in the search direction).

V. NUMERICAL EXPERIMENTS

The numerical examples we discuss here are meant to suggest various trends and possibilities that may occur in practice in the context of optimization when the two different MCF strategies are implemented. It is clear that the optimality question of one strategy versus another is very difficult, because the overall effect on minimization is cumulative. For this reason, general conclusions may only be reached after thorough investigations on a wide variety of problems. Nonetheless, our current experience may already prove useful to investigators and we share it in hope that further studies will be motivated.

We study in detail the effect of the different MCF strategies for three types of objective functions with variations of size, starting points, parameters, and preconditioners. We use TNPACK with the GMW and S&E factorizations implemented as discussed previously.

Three versions of MCF were implemented: GMW; S&E (a) — with the condition $e_j \geq e_{j-1}$ imposed; and S&E (b) — with this condition omitted. The tolerance parameters were set to $\tau_1 = \tau_2 = 1.0\text{E}{-}05$ for S&E, and $\tau = 1.0\text{E}{-}08 * \max\{1, \gamma, \xi\}$ for GMW. This value of τ for GMW was chosen since our typical value for γ ($\gamma > \xi$) in the computational chemistry problems has magnitude of $O(10^3)$. The convergence criterion to reach a minimum was set to $\|\mathbf{g}\| \leq 1.0\text{E}{-}08 * \max\{1, \|\mathbf{x}\|\}$ where $\|\cdot\|$ denotes the standard Euclidean norm.

Two functions are chosen from computational chemistry: one representing a typical case of a molecule where distinct and well-separated conformational regions exist (due to steric hindrance), and one of a cluster of molecules, possessing a very large number of low-energy configurations — which increase with size — due to a complex continuum of architectural possibilities. For both examples, natural preconditioners arise from the separability of the energy function into local and nonlocal forms (details follow) [12-14]. Our third problem is the trigonometric function [15], appropriate here due to the following characteristics: the Hessian and preconditioner are indefinite at the standard starting point ($\mathbf{x}_0$) and remain so for many iterations right until a convex region is entered; there are many negative *and* clustered eigenvalues near $\mathbf{x}_0$; minimization becomes more difficult with increasing size; and, since the Hessian is dense, it is natrual to consider preconditioners other than diagonals. It is worth pointing out that for examination of the MCF in our context — indefinite preconditioners in the inner CG loop — "standard" test problems present a difficulty because natural, much less indefinite, preconditioners are nontrivial to formulate. Our experience has been that diagonal preconditioners typically work best; unfortunately, diagonal matrices are poor choices for our study because of their simplicity and the consequence that the main feature of S&E — looking ahead in the factorization for possible small diagonal candidates — is not tested. In future work, we will examine the issue of preconditioners for standard test problems in more detail so that another MCF comparison study can be performed.

Efficiency of overall convergence in our truncated Newton method can be determined

from the total number of Newton iterations, PCG iterations, and function & gradient evaluations. The total cost of the method depends on these sums as well as the cost of solving the sparse system $M\mathbf{z} = \mathbf{r}$ in PCG. The cost for this system — solved by our sparse MCF version — depends on the sparsity structure of M and can be as low as $O(n)$ [3]. The number of function & gradient evaluations is at least one per Newton iteration, more than one if required by the line search. Overall, for truncated Newton methods, the *total* number of PCG iterations is generally a good measure for performance comparison. The total number of Newton iterations provides additional information on overall progress since many Newton iterations typically indicate that indefiniteness (of H) has been detected, in which case the inner loop is terminated without satisfaction of the truncation criterion.

It is important to realize that the effect of using different MCF procedures for indefinite preconditioners in combination with truncation of the inner loop leads to different search directions. Note that even if the truncation criterion becomes very strict (i.e., very small δ in eq. (16)), the inner loop may be halted if H is indefinite with detection of a negative-curvature direction. In this case, different search directions can also result. Thus, the effect of the MCF on the overall minimization progress is indirect and cumulative. Furthermore, in addition to affecting the total number of Newton iterations, PCG iterations, and function & gradient evaluations, different MCF strategies may produce different minima if several local minima exist. This is, in fact, our experience with computational chemistry problems.

For large-scale problems, the number of PCG iterations is often limited to a number $\ll n$ due to the prohibitive cost of the inner iteration. Thus, while the effect of a particular MCF strategy in one Newton step is interesting, more important from a practical point of view is comparing the overall effect on minimization progress and performance. Nonetheless, in the examples below, we provide the total number of PCG iterations for each minimization run as a cumulative indicator of the first question (the step-by-step effect). In all our examples, similar performance trends were also observed when the truncation criterion was set to be very strict, to force small residuals for the Newton equations

and thereby mimic exact Newton methods. For the first molecular model, we provide the data for such a run alongside the standard 'truncated' version.

All computations were performed in double precision on a DEC Vax-3600 (3-processor) computer at the Courant Institute of Mathematical Sciences, New York University. Machine precision is of order 10^{-17}. For the present work, we have chosen to study problems of moderate sizes to permit a thorough investigation of the MCF segment in minimization, including eigenvalue analysis of H and M at every iteration.

(A) A Nucleic Acid Component

In the "static" semi-empirical approach of computational chemistry, a target potential energy function is minimized to find three-dimensional structures of biomolecules in low energy configurations. Potential energy surfaces are typically very complex, involving many local minima, maxima, and transition points. Indeed, this problem structure makes the MCF component in optimization particularly important. Our main area of interest involves nucleic acids, and the first example described here involves a component of DNA, deoxycytidine (dC), which we have studied in detail [12-13]. Biochemical details of our molecular mechanics and dynamics package, MADPAC, have been discussed elsewhere [12-14]. We mention here only that, for this model, potential energy parameters have been well tested and specific local minima are known experimentally.

Molecular preconditioners used in the MCF segment of TNPACK consist of the *local* chemical interactions between bonds, bond angles, and dihedral angles. Such contributions tend to cluster near the diagonal when neighboring atoms are numbered sequentially. More recently, penalty terms for rigid-body rotations and translations were also added to our preconditioner (this eliminates corresponding zero eigenvalues at the solution). We have found this choice of M to be a convenient sparse approximation to the (dense) Hessian that works well in practice. However, there is no guarantee that M is positive definite, even near a local minimum. Thus, the overall effect on minimization of different MCF

methods will be particularly interesting to investigate.

In Tables I-IV, we show results for the dC model involving 87 variables. Tables I and II show the sets of results from 3 starting points (**x1**, **x2**, **x3**): in Table I with the truncation parameter $\delta = 1$ (eq. (16)), and in Table II with $\delta = 1.0\text{E}{-}08$. Tables III and IV provide eigenvalue information on H and M at the starting points and at 3 computed solution points (**x5**, **x6**, **x7**). In Tables I and II we show the total number of Newton iterations, PCG iterations, and function & gradient evaluations for each run. Each run from the **x1**, **x2** and **x3** group converged to the local minima **x6**, **x5**, and **x7**, respectively. The MCF information per run is given by the minimum, maximum, and median values for $\|E\|_\infty$, computed over all Newton iterations. Additionally, the number of Newton iterations for which $\|E\|_\infty > 0$ is indicated.

For dC minimization, many starting conformations were investigated, and the 3 different choices shown here are representative of different energies and eigenvalue distributions at the initial points. In energy ranking, $F(\mathbf{x1}) < F(\mathbf{x2}) < F(\mathbf{x3})$ with corresponding energy values of 1.4, 1.8E+03, and 5.1E+04 Kcal/mol. All Hessians and preconditioners are indefinite at the starting points. From the values of λ_{min}, λ_{max}, and the number of negative eigenvalues (#NEG) shown in Tables III and IV, we see that while M has 1 negative eigenvalue of magnitude $O(10^{-1})$ and a λ_{max} of $O(10^3)$, the distribution of H's eigenvalues differs at the 3 starting points. At **x1**, H has 2 negative eigenvalues with λ_{min} of magnitude $O(10^0)$ and a λ_{max} of $O(10^3)$; at **x2**, H has 7 negative eigenvalues with λ_{min} of magnitude $O(10^4)$ and λ_{max} of $O(10^5)$; and at **x3**, H has 9 negative eigenvalues with λ_{min} of magnitude $O(10^5)$ and λ_{max} of $O(10^7)$. As for the minima (**x5**, **x6**, **x7**), both H and M are positive definite with reasonable condition numbers of $O(10^4\text{-}10^5)$. In energies, $F(\mathbf{x5}) < F(\mathbf{x6}) < F(\mathbf{x7})$ with values of $-6.8, -5.7$, and -5.0 Kcal/mol, respectively. These energy differences are large in molecular terms and separate well 3 different geometric configurations.

From Table I, in which the truncation parameter was large ($\delta = 1$), we first note

that overall convergence to a local minimum is very rapid with the truncated Newton method: $< n/4$ outer iterations, $< n$ inner iterations, and $\sim n/2$ function & gradient evaluations. Second, in the MCF segment, we note that M is modified in only 1 (the first) and possibly another (typically, the second) iteration. The values of $\|E\|_\infty$ are larger for GMW than for S&E, by about a factor of 2, but in both strategies $\|E\|_\infty$ is greater than λ_{min} by 1 or 2 orders of magnitude. When the MCF produces nonzero increments in only one Newton iteration, $\|E\|_\infty$ corresponds to λ_{min} of M at the starting point (shown in Table III). When there are two iterations with M indefinite, the maximum recorded value for $\|E\|_\infty$ corresponds to λ_{min} of M at the starting point, while the recorded minimum $\|E\|_\infty$ corresponds to a second λ_{min} which is slightly smaller in magnitude than the first ($\lambda_{min} = -0.22$ vs. $\|E\|_\infty = 2.4$ in the second iteration of S&E (a) for **x1**, and $\lambda_{min} = -0.01$ vs. $\|E\|_\infty = 6.1$ in the second iteration of S&E (a) for **x3**). The values of γ and ζ (eqns. (6) and (7)) during the minimization runs are typically around 2000 and 1000, respectively. In the first Newton iteration when H is indefinite, between 1-3 PCG iterations are performed in all runs. The smaller S&E increments do not produce a systematic reduction of PCG iterations in this context. In all MCF segments, diagonal increments are only made in the last 3-4 steps ($j \geq 83$).

The overall trend to be noted here is that despite the fact that $\|E\|_\infty$ is systematically greater for GMW than for S&E, both strategies perform overall quite similarly. The differences are small, but the GMW version appears slightly more efficient. Among the two S&E versions — with and without the condition $e_j \geq e_{j-1}$ imposed — performance differences are also small. Close analysis of the data shows that relaxing the restriction that $e_j \geq e_{j-1}$ can lead to slightly greater $\|E\|_\infty$ values per Newton iteration but that $e_j = 0$ for more diagonal components. Thus, small differences in performance depend on the differences in effective preconditioners $(M + E)$ of H. Since nonzero increments are only involved in the last few steps, overall performance between all versions is very similar.

When the truncation parameter was made more strict (δ = 1.0E−08), we note from

the corresponding set of runs shown in Table II that overall trends of the S&E and GMW versions are repeated. This behavior verifies that the cumulative effects we reported above reflect systematic differences in the MCF strategies (per step) rather than just different resulting paths. All MCF increment values and trends are very similar to those found in Table I. The power of truncation can also be clearly noted: overall convergence is not sacrificed by approximating the search directions in early steps; although the number of Newton iterations may be slightly less in all Table II runs, the required number of PCG iterations is greater by a factor of 3-4 than the corresponding values of Table I.

Overall, this rapid convergence and insensitivity of the minimization to the MCF strategy may be attributed to the fact that this optimization problem is straightforward, with distinct convex conformational regions. Indeed, in almost all runs, H becomes positive definite after the first Newton iteration and remains so. Moreover, M appears to be a good preconditioner for this problem, and it is positive definite except during 1 or 2 Newton iterations. More important for the MCF details, near $\mathbf{x}_0$ M has only 1 *isolated* negative eigenvalue; all MCF strategies modify only a few diagonals and are all successful at "fixing" the problem efficiently and rapidly.

As for the magnitude of $\|E\|_\infty$, these runs further support the result that $\|E\|_\infty$ may be smaller in practice for S&E than for GMW. However, they also suggest that this factor alone does not imply better performance in the context of optimization. In fact, larger modifications to M may result with GMW but may also produce very good descent directions, leading to more efficient minimization progress.

(B) Water Clusters

Our second model of water clusters differs from the dC model in two major respects. First, while there are distinct energy minima for dC that are well separated by high-energy regions, liquid-water clusters possess many nearby local minima that are close in energy [14]. This results from the fact that the clusters are stabilized by a random and

rich intermolecular network of hydrogen bonds. As the cluster size grows, the number of possible hydrogen-bonded configurations increases enormously. Second, as a result of this difference in potential-energy surface, the Hessian for water, unlike dC, is often indefinite. In most of our runs, the inner loop terminates when a direction of negative curvature is detected; this occurs for many Newton iterations until the last few, when some convex region is entered. This behavior typically results in a large number of inner and outer iterations as well as function & gradient evaluations (the latter, presumably to reduce the energies sufficiently when poor search directions are produced). Furthermore, our "local potential" preconditioner for water does not contain the major energy contributions as in the dC model. The non-local terms, in particular the electrostatic potential which leads to appropriate formation of hydrogen bonds, is clearly dominant. Thus, the MCF often produces nonzero modifications at every Newton iteration. Since overall performance of the truncated Newton method depends on the convexity of the problem and on the choice of M, finding optimal cluster configurations is a much more difficult problem than finding the minima of dC. The added difficulty of choosing starting points compounds the problem.

Results for a water dimer model (2 water molecules, 3 atoms per molecule, $n = 18$) are summarized in Table V. There is only one minimum for the dimer, corresponding to a linear hydrogen-bonded pair. In general, convergence requires around 40 Newton iterations. Run A used our local M. For this run, GMW is more efficient in terms of total Newton and PCG iterations as well as function & gradient evaluations. The differences in the $\|E\|_\infty$ values are interesting. GMW produces very small diagonal elements at the end of the factorization — $O(10^{-13}\text{-}10^{-20})$ — and consequently leads to increments in the order of the prescribed parameter τ (see eq. (9)). The S&E versions, on the other hand, switch to Phase 2 earlier in the factorization, thereby leading to more diagonal modifications; nevertheless, this avoids formation of very small diagonal entries. To illustrate, we show in Table VI-(A) the diagonal modifications for all three MCF versions in the first Newton iteration. Similar behavior occurs in all other Newton iterations.

To examine these emerging trends more closely, we ran the same minimization problem with a different choice of M: some of the nonlocal terms (intramolecular oxygen–oxygen interactions) were added to M of run A. Essentially, we added a small off-diagonal (3×3) block to the previous block-diagonal M made of two 9×9 blocks. Results, summarized in Table V under run B, show that *overall* performance of all three MCF versions is now very similar. However, the mean $\|E\|_\infty$ from GMW is greater than that of the S&E versions by about 1 order of magnitude. As in run A, GMW begins diagonal increments later than S&E, and this leads to larger increments. This behavior is illustrated in Table VI-(B).

A systematic investigation was then performed for water clusters of increasing size. Starting points were chosen by positioning the molecules within a computational box, specified by the number of molecules along the x, y, and z directions: each molecular oxygen is positioned at a center of a box, with its hydrogens lying on a unit sphere around it (their positions are chosen pseudorandomly), and each molecule is separated from its neighbors by some chosen distance [14]. Such configurations are typically high in energy and very far from optimal structures; in particular, large clusters adopt a spherical, rather than rectangular arrangement, so large structural distortions are involved during minimization. In addition, different cluster sizes possess varying numbers of geometric arrangements, so while a 3x3x3 molecular cluster (27 molecules, $n = 243$) adopts an overall spherical arrangement, a 2x5x2 ($n = 90$) cluster can adopt a large number of different shapes. This behavior, in combination with the random aspect of selecting starting points, leads to differences in minimization performance that are not always correlated monotonically to the cluster size. The limiting number of PCG iterations (per Newton iteration) was set to n in all runs to make the MCF comparison as fair as possible. Our truncation parameter δ was set to unity.

Table VII compares performance among our MCF versions for water clusters of dimensions $n = 27$ to $n = 576$, and Table VIII provides eigenvalue information for the corresponding starting and final points and for the water dimer. In the runs we report for

each dimension, the resulting minima are very close in energy although they may not be identical. In such cases, corresponding eigenvalue information at the final points is very similar for each group, and the values we report in Table VIII are representative.

In comparison with results for the water dimer discussed above, similar behavior in the $\|E\|_\infty$ size can be observed for larger water models, but a new overall trend emerges among the MCF versions with increasing size. We note from Table VII that the GMW version is more efficient for smaller dimensions. However, as the size increases, the S&E strategy — in particular the S&E(b) variation — becomes more efficient overall. In particular, differences in total number of PCG iterations between the S&E and GMW versions become large in many cases.

Close analysis of the data reveals the following explanation. The Hessian is indefinite at the starting point and remains so for many iterations; thus, the inner loop often ends with detection of negative curvature. M is also indefinite, with approximately $n/9$ negative eigenvalues (all very close to zero) throughout the run, forcing nonzero modifications at every step of minimization. Now, the GMW version produces $\|E\|_\infty$ sizes on the order of the set tolerance τ ($O(10^{-5})$), while the Gerschgorin estimates of S&E lead to much larger modifications ($O(10^3)$). Indeed, behavior in larger water clusters is very similar to the situation shown in Table VI-(A) for the water dimer. Thus, the resulting GMW search directions often produce very slow progress for many iterations, and the number of H's negative eigenvalues decreases slowly; when H becomes positive definite, the truncation criterion is stricter (since k is larger, see eq. (16)) and many PCG iterations are then involved. In fact, the last few iterations of GMW for large dimensions often end when the maximum number of allowed PCG iterations has been reached. The S&E strategy, on the other hand, succeeds in entering positive definite regions of H more quickly, and progress is typically rapid thereafter. The inner loop then terminates successfully with satisfaction of the permitted residual (rather than excess of PCG iterations), although the number of PCG iterations involved may be large. Version (b) of S&E is more competitive than version

(a) because diagonal modifications are made less often, so a smaller number of total PCG iterations typically results. When we performed the same runs with a stricter truncation criterion (as in the dC example), results were very similar to those reported above since a stricter criterion affects only iterations that do not end in detection of negative curvature, and these iterations only occur at the end, where already the Newton equations are solved more accurately.

Why is the GMW version competitive for smaller dimensions? The number of negative eigenvalues seems to play a key role in steering the minimization progress. Note from Table VIII that H has about $n/3$ negative eigenvalues at the starting point, with a λ_{min} of magnitude $O(10^0\text{-}10^1)$ and a λ_{max} of $O(10^3)$. The number of M's eigenvalues is smaller and about $n/9$, but they are all very small in magnitude, clustered, and persistent. In fact, the existence of zero eigenvalues for M results from the interdependence of the potential energy derivatives in the x, y, and z directions for terms associated with each water molecule. Thus, when only a few diagonal candidates are small (say, 1-3), GMW is more efficient because the minimal amount of modification is sufficient and does not affect the computed search direction dramatically. However, as the number of small diagonals increases, slow progress results from many more resulting diagonals of magnitude τ, 10^{-5}. The S&E strategy detects indefiniteness early in the factorization (Phase 1 is typically false very early in the Cholesky factorization, after about 10 steps) and modifies the diagonals more substantially. This preventive strategy "pays off" in overall performance as the size and complexity of the problem increase.

These results suggest that resetting very small prescribed tolerance parameters can slow minimization progress considerably. This situation can occur for problems with many near-zero eigenvalues, a typical one in computational chemistry because of translation and rotation invariance as well as dependence of the potential-energy derivative components. They further suggest that the S&E strategy may be very effective for handling such complex large-scale problems.

(C) The Trigonometric Function

To examine the preventive S&E strategy and resulting modifications further, we focus on a different problem structure where many negative eigenvalues of larger magnitudes are present. The trigonometric function of variable dimension n is given as [15]:

$$F(\mathbf{x}) = \sum_{j=1,\ldots,n} \left(n - \sum_{i=1}^{n} \cos x_i + j(1 - \cos x_j) - \sin x_j \right)^2 . \tag{19}$$

The recommended starting point is $\mathbf{x}_0 = (1/n, \ldots, 1/n)^T$, at which H is indefinite. We study 4 sets of minimization runs from $\mathbf{x}_0$, for $n = 50$, 100, 250, and 1000, and for each set we use diagonal, tridiagonal, and pentadiagonal preconditioners, constructed from the Hessian. We set our truncation parameter δ to unity and the limiting number of PCG iterations (per Newton step) to 10. This limit is imposed because H is dense so the inner iterations are costly. Typically, the residual is very small after a small number of PCG iterations, and this limit has only a small effect, if any, on the last 1-2 Newton iterations. Additionally, since the residual becomes very small quickly, a stricter truncation criterion has little effect on performance. For all runs, we use a stricter overall minimization termination criterion of $\|\mathbf{g}\|/\sqrt{n} \leq 1.0\text{E}{-}12 * \max\{1, \|\mathbf{x}\|\}$, since $\|\mathbf{g}(\mathbf{x}_0)\|$ is small and we wanted to ensure that a region of quadratic convergence is well entered. When such a region is entered, the gradient is reduced very rapidly, so this strict condition does not present a problem.

In Table IX we show results for the 4 sets of runs, and in Table X we list the eigenvalues of H and the pentadiagonal M at both $\mathbf{x}_0$ and $\mathbf{x}_*$ (solution). For the other choices of M — diagonal and tridiagonal — only small changes for λ_{min} were noted ($\sim$ 0.01 - 0.05). The typical values of γ and ξ for this function are $\gamma \sim 2.0$ and ξ in the range O(10^{-2}-10^{-1}). From the eigenvalue table we note immediately that at $\mathbf{x}_0$ both H and M have 60% of their eigenvalues negative. Moreover, our eigenvalue analysis shows that all eigenvalues have magnitudes O(10^{-1}-10^{-2}), most of them of O(10^{-1}). For example, for $n = 100$, the 61 negative eigenvalues of the pentadiagonal M occur in increasing order of {-0.591,

$-0.591, -0.584, -0.584, -0.577, -0.577, -0.571, -0.571, -0.570, -0.565, -0.561, -0.559, -0.556, -0.552, \ldots, -0.140, -0.112, -0.069, -0.045, -0.019\}$. At $\mathbf{x}_0$, λ_{max} is about 1.5 for H and M, so the eigenvalues are concentrated in the $[\ -1.0, 1.5\ ]$ interval. At $\mathbf{x}_*$, H and M are positive definite, with λ_{min} of $O(10^{-2})$ and λ_{max} of about 3.0 for H and 2.0 for M.

From the runs with *diagonal* M we note from Table IX that the GMW strategy is the most effective for each dimension. The required number of Newton iterations is considerably less than for the S&E runs (except for $n = 50$, where it is the same), and the total number of PCG iterations is smaller. The number of iterations involving nonzero modifications to M is also smaller for GMW for $n > 50$ (equal to S&E for $n = 50$). Note that overall superiority of GMW occurs even though the size of $\|E\|_\infty$ is typically greater than that of S&E, by a factor of 2. This relation is expected from the modification formulas (eqs. (9) and (13)) since $\theta = 0$ and τ is greater than the negative candidate diagonal. Thus, while GMW generally modifies negative diagonals m_{jj} to $|m_{jj}|$, S&E produces elements of magnitude τ. These small diagonals produce search directions that lead to slower progress during minimization for S&E than for GMW (further discussion is given below).

We note an interesting trend when off-diagonal elements are added to M. In short, the relative performance of GMW is now slowed — more Newton iterations, PCG iterations, and nonzero modifications — while performance of S&E is greatly improved. In particular, as the size of the problem increases, version S&E(a) becomes superior. Thus, the preventive strategy of S&E works very well for problems with this eigenvalue structure, as it is very effective at getting out of indefinite regions more quickly. The small number of Newton iterations, especially for S&E(a), results since H and M become positive definite quickly, and rapid progress is made thereafter.

This difference in progress can be seen from Figure 2, which shows λ_{min} vs. $\|E\|_\infty$ for the 3 MCF versions for the run with $n = 250$ with tridiagonal M. During the first 5

Newton iterations, the number of negative eigenvalues of M in the GMW version is reduced slowly from 152; the GMW sequence of negative eigenvalues per Newton iteration starting from iteration 0 is $\{152, 135, 102, 44, 55, 31\}$, while the corresponding S&E(a) and S&E(b) sequences are $\{152, 5, 2, 0, 0, 0\}$ and $\{152, 5, 4, 0, 0, 0\}$, respectively. At later iterations, the number of negative eigenvalues in the GMW run ranges from 0 to 65, while the range for the S&E versions is between 0 and 8. An examination of the data shows that the first Newton iteration is quite similar for all MCF versions in the modification schedule: GMW modifies diagonals $99 \leq j \leq 250$, and S&E enters Phase 2 at $j = 97$. The GMW and S&E versions perform 2 and 1 PCG iterations, respectively, at this first Newton iteration until a direction of negative curvature is detected. Now, at the second Newton iteration, GMW modifies negative diagonal candidates in the range $96 \leq j \leq 250$, of magnitude range $O(10^{-1}$-$10^{-3})$, while both S&E versions enter Phase 2 at $j = 153$ and modify few small negative diagonals *and* small positive diagonal candidates of $O(10^{-2}$-$10^{-4})$; consequently, future diagonals do not become negative. This trend between the 3 MCF versions repeats for several Newton iterations, involving less and less diagonal modifications at each iteration, and progress is particularly rapid for S&E(a). Version S&E(b) is slower than S&E(a) because many resulting increments are of order τ when the condition $e_j \geq e_{j-1}$ is omitted, and smaller diagonal elements appear in later Cholesky steps.

Thus, the tendency of small diagonal candidates to form is reversed between the GMW and S&E versions for our diagonal and non-diagonal M runs. Note, however, that this tendency is overcome in each version by a different strategy — larger GMW increments in the diagonal case, and the S&E preventive schedule in the non-diagonal cases. The outcomes in each case explain our noted differences in performance among the 3 MCF strategies for the trigonometric function. Relating this to our previous examples, we recall that for the water clusters, a systematic trend of small, positive diagonals of order τ also slowed overall minimization progress. Together, these results further support the great effectiveness of the S&E strategy for general indefinite matrices with negative eigenvalues

that have magnitudes greater than τ. The clustering of eigenvalues also seems to make the S&E strategy more effective, because the preventive modification schedule improves performance later on. Interestingly, it has already been noted that S&E produces the most significant improvement in $\|E\|_\infty$ over GMW for problems where the eigenvalues are clustered between -1 and 1 [1].

VI. SUMMARY

We have discussed the implementation of the new modified Cholesky factorization (MCF) of Schnabel & Eskow (S&E) in the context of large-scale nonlinear optimization. A version of the factorization was implemented in our optimization program TNPACK, a large-scale truncated-Newton package. The MCF is used in the Preconditioned Conjugate Gradient component (PCG) to solve a linear system involving the preconditioning matrix of the Newton equations. The preconditioner is often chosen from the physics of the problem but is not guaranteed to be positive definite. Both the S&E and the Gill, Murray, and Wright (GMW) factorizations were incorporated into TNPACK, and performance comparisons were made for sample problems of different eigenvalue distributions.

Two areas have been discussed in this paper: implementation in large-scale optimization, and performance results. The no-pivoting version of S&E is used because large systems, when subjected to direct factorizations, are typically sparse. Thus, reordering is often done a priori. Moreover, in many physical applications, sparsity patterns are structured and pivoting is not done at all, since it may destroy optimal patterns.

Implementation of S&E differs from GMW both in the form of the recipe that determines the diagonal increments and in the manner by which the recipe is activated. While GMW subjects all candidate diagonals to the same modification formula, S&E employs a two-phase strategy. Phase 1 denotes a state when all diagonal increments are zero (positive definite matrices are always in Phase 1), and Phase 2 is entered when any prospective diagonal in the remaining submatrix is small. Only at Phase 2 is the Gerschgorin-based

recipe of S&E applied. Whereas the phase test in S&E requires that the diagonal elements of the matrix be updated at every step, this is not a requirement in GMW. For the actual recipe of S&E, we suggest two variations that may work better in certain optimization applications: 1) e_j is not required to be as large as e_{j-1}, and 2) the same strategy is used for all the e_j's (instead of using actual eigenvalues in the last two steps). We used the same strategy for the e_j's throughout this work, and we compared two S&E versions where $e_j \geq e_{j-1}$ was enforced (S&E,a) and where it was not (S&E,b).

In the numerical experiments, we focused on the *overall* efficiency and performance of minimization when different MCF strategies are used to produce positive definite preconditioners. Clearly, the overall performance is the important aspect in practice, and it is certainly of major importance when the number of PCG iterations is limited to a small number due to computational cost of the problem. Although the effect of different MCF methods is cumulative, systematic trends have emerged in this work. In this respect, the truncated Newton formulation is a particularly interesting vehicle to test these different MCF strategies because the cumulative effects are easier to observe. An examination of analogous runs where the truncation criterion was set to be more strict exhibited the same overall performance trends.

The three problems we examined in detail, by varying size, preconditioners, starting points, and truncation parameters, involve three different eigenvalue distributions that are typically encountered in practice: an isolated negative eigenvalue, several very small-in-magnitude (near zero) positive and negative eigenvalues, and a large percentage of negative eigenvalues. In the first case, represented by a molecular model with distinct and well-separated conformational minima, nonzero diagonal modifications to M were required in only 1 or 2 Newton iterations. The GMW and S&E factorizations behaved very similarly, though GMW was slightly more efficient despite the fact that $\|E\|_\infty$ was typically higher by a factor of 2. In the second case, a molecular-cluster model possessing numerous and clustered minima and a characteristically indefinite potential surface, nonzero modifica-

tions were made in all iterations. Now GMW was more efficient for smaller dimensions but S&E(b) became much better as size increased; $\|E\|_\infty$ was typically around the set tolerance τ, $O(10^{-5})$, for GMW and of $O(10^2\text{-}10^3)$ for the S&E versions. In the third case of the trigonometric function, nonzero modifications were involved at some Newton iterations. GMW performed better for diagonal matrices of various dimensions, but S&E(a) became superior as off-diagonals were added. $\|E\|_\infty$ was typically higher for GMW by a factor of 2 or more. Our results have shown that the relative performance between the GMW and S&E factorizations depends on such different problem characteristics. Our observations can be summarized as follows:

(1) S&E may indeed lead to smaller values of $\|E\|_\infty$ than GMW, as suggested by theoretical estimates. Nonetheless, the size of $\|E\|_\infty$ may not be the crucial factor in the context of optimization; the modification schedule and its consequences are also very important for maintaining numerical stability and directing progress. Thus, GMW can be more efficient in some cases, even when $\|E\|_\infty$ is larger.

(2) The two-phase strategy of S&E clearly leads to earlier modifications of the diagonal elements than GMW. While causing more perturbations to the original matrix, the S&E phase-switch strategy nonetheless avoids formation of more negative elements that tend to appear with GMW later in the factorization. This strategy can be very effective at getting out of indefinite regions, especially in cases when there are many near-zero eigenvalues.

(3) The two variations we suggested for the S&E increment strategy work well when applied to large, sparse linear systems in the context of optimization. The special formula for e_n and e_{n-1} may be unnatural for problems with inherent symmetry, and implementing the requirement that $e_j \geq e_{j-1}$ can lead to unnecessary perturbations of the original matrix. However, this requirement may be more effective for problems with many negative eigenvalues as a means of accelerating progress.

It appears that the basic question with the S&E factorization versus GMW in the

context of optimization is: does the preventive strategy of S&E "pay off"? For problems with very few and isolated negative eigenvalues, the GMW strategy seems to be more effective: it remedies the problem only as necessary. The S&E strategy, on the other hand, may rely on large Gerschgorin estimates and thus may lead to unnecessary modifications which do not accelerate overall progress. However, for problems characterized by highly indefinite regions, especially with clustered negative eigenvalues, the S&E strategy is very effective. The probability of encountering many negative diagonals is high, so the schedule and size of the modifications are critical. In particular, as the size and complexity of the problem increases, the relative superiority of the S&E increases because by remedying the problem earlier, S&E leads to better numerical behavior during the factorization (i.e., no overflow/underflow).

These results suggest the following classification of cases where the two methods will be effective. The GMW technique may be more effective for minimization of functions that are well approximated by local quadratic models in the region of interest. In this case, H — and M in our truncated Newton context — are likely to be close approximations to a positive definite matrix. The S&E strategy may work better for large, indefinite systems, especially with many clustered negative eigenvalues, where diagonal candidates tend to grow more negative if modifications begin too late. This may occur in minimization for starting points that are far away from local minima; for functions that possess many local minima, as in computational chemistry applications; and, in general, for highly nonlinear objective functions. Indeed, the local convexity of the model has been suggested as an important factor in evaluating performance of the truncated Newton method in relation to other large-scale optimization methods [17]. Our results suggest that incorporation of the S&E modified Cholesky factorization in minimization of highly nonlinear functions may greatly improve the performance and competitiveness of Newton methods.

REFERENCES AND FOOTNOTES

[1] R. B. Schnabel and E. Eskow, "A New Modified Cholesky Factorization", Computer Science Department Technical Report CU-CS-415-88, University of Colorado, Boulder, Colorado (1988).

[2] E. Eskow and R. B. Schnabel, "Software for a New Modified Cholesky Factorization", Computer Science Department Technical Report CU-CS-443-89, University of Colorado, Boulder, CO (1989).

[3] T. Schlick and A. Fogelson, "TNPACK — A Truncated Newton Minimization Package for Large-Scale Problems", submitted.

[4] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, pp. 105-111, Academic Press, London (1983).

[5] R. S. Dembo and T. Steihaug, "Truncated-Newton Algorithms for Large-Scale Unconstrained Optimization", *Math. Prog.* **26**, 190-212 (1983).

[6] S. G. Nash, "Preconditioning of Truncated-Newton Methods", *SIAM J. Sci. Stat. Comput.* **6**, 599-616 (1985).

[7] T. Schlick and M. Overton, "A Powerful Truncated Newton Method for Potential Energy Minimization", *J. Comput. Chem.* **8**, 1025-1039 (1987).

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Second Edition, Johns Hopkins University Press, Baltimore, MD (1989).

[9] S. C. Eisenstat, M. C. Gursky, M. H. Schultz, and A. H. Sherman, "Yale Sparse Matrix Package, I. The Symmetric Codes", *Intl. J. Num. Met. Eng.* **18**, 1145-1151 (1982).

[10] S. C. Eisenstat, M. H. Schultz, and A. H. Sherman, "Efficient Implementation of Sparse Symmetric Gaussian Elimination", in *Advances in Computer Methods for Partial Differential Equations*, 33-39, ed. R. Vichnevetsky, AICA, New Brunswick, New Jersey (1975).

[11] S. C. Eisenstat, M. H. Schultz, and A. H. Sherman, "Algorithms and Data Structures for Sparse Symmetric Gaussian Elimination", *SIAM J. Sci. Stat. Comp.* **2**, 225-237

(1981).

[12] T. Schlick, B. E. Hingerty, C. S. Peskin, M. L. Overton, and S. Broyde, "Search Strategies, Minimization Algorithms, and Molecular Dynamics Simulations for Exploring Conformational Spaces of Nucleic Acids", *Theoretical Biochemistry and Molecular Biophysics: A Comprehensive Survey*, ed. D. L. Beveridge and R. Lavery, Adenine Press, Guilderland, New York, in press.

[13] T. Schlick, "Modeling and Minimization Techniques for Predicting Three-Dimensional Structures of Large Biological Molecules", Ph.D. Thesis, New York University, Department of Mathematics (1987). Available through University Microfilm International.

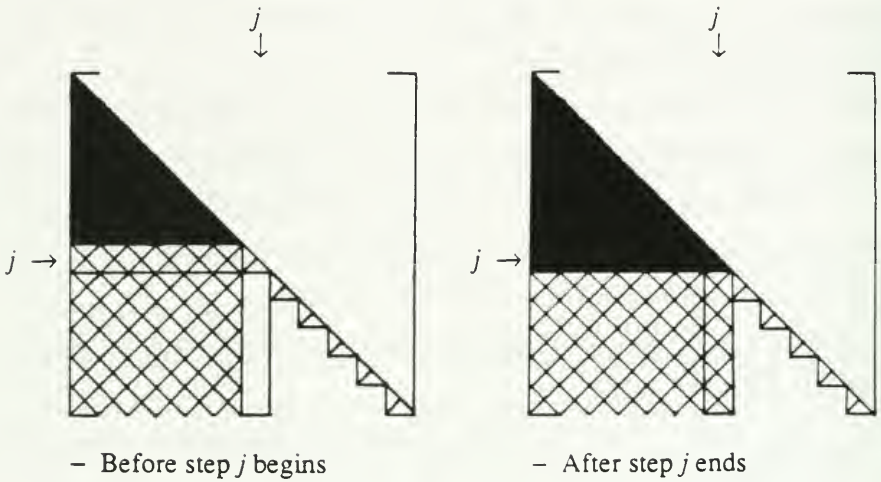
[14] T. Schlick, S. Figueroa, and M. Mezei, "A Molecular Dynamics Simulation of a Water Droplet by the Implicit-Euler/Langevin Scheme", submitted.

[15] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "Testing Unconstrained Optimization", *ACM TOMS* **7**, 17-41 (1981).

[16] Some earlier runs with a different choice of preconditioner did display this latter type of behavior. In GMW, M was indefinite at the first Newton iteration and, after a very large $\|E\|_\infty$, remained positive definite throughout the run; in the analogous S&E run, M was modified by a much smaller value at the first iteration, but this led to nonzero modifications at every iteration. This behavior was not repeated in the second set of runs with the strict truncation criterion, because equally good search directions were produced.

[17] S. G. Nash and J. Nocedal, "A Numerical Study of the Limited Memory BFGS Method and the Truncated-Newton Method for Large-Scale Optimization", Technical Report NAM 02, Dept. of Electrical Engineering and Computer Science, Northwestern University, Illinois.

Figure 1: *Step j of the Modified Cholesky Factorization*



■ contains final L and D factors

⊠ contains the auxiliary quantities of LD

⊘ contains updated diagonal elements

Figure 2: λ_{min} *vs.* $||E||_\infty$ *for the Trigonometric Function*

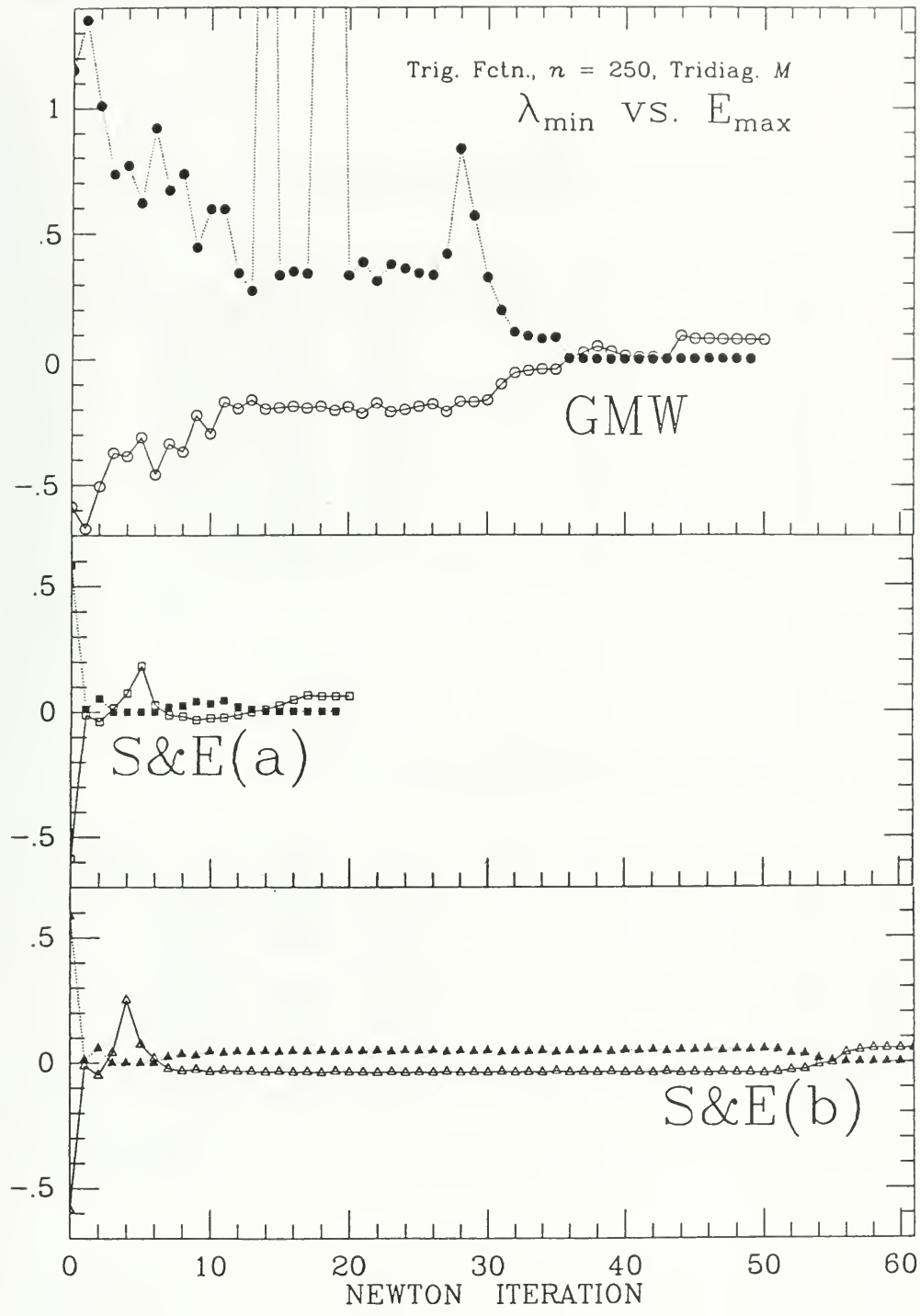


Table I

Molecular Model 1: Deoxycytidine (dC), n = 87
(δ = 1.0 in truncation criterion)

	Run	Newton Itns.	PCG Itns.	F&G Evals.	$\|\|E\|\|_\infty$ Min.	Max.	Med.	Nonzero Itns.
(X1)	GMW	15	71	38	1.3E+01	1.3E+01	1.3E+01	1
	S&E,(a)	17	84	43	2.4E+00	5.6E+00	4.0E+00	2
	S&E,(b)	16	81	41	5.6E+00	5.6E+00	5.6E+00	1
(X2)	GMW	19	79	48	1.4E+01	1.4E+01	1.4E+01	1
	S&E,(a)	21	79	50	7.2E+00	7.2E+00	7.2E+00	1
	S&E,(b)	19	86	46	7.2E+00	7.2E+00	7.2E+00	1
(X3)	GMW	13	48	28	1.3E+01	1.3E+01	1.3E+01	1
	S&E,(a)	13	49	31	6.1E+00	7.2E+00	6.7E+00	2
	S&E,(b)	14	57	27	7.2E+00	7.2E+00	7.2E+00	1

Table II

Molecular Model 1: Deoxycytidine (dC), n = 87
(δ = 1.0E-08 in truncation criterion)

	Run	Newton Itns.	PCG Itns.	F&G Evals.	$\|\|E\|\|_\infty$ Min.	Max.	Med.	Nonzero Itns.
(X1)	GMW	12	207	29	1.3E+01	1.3E+01	1.3E+01	1
	S&E,(a)	13	230	31	5.6E+00	5.6E+00	5.6E+00	1
	S&E,(b)	12	191	29	5.6E+00	5.6E+00	5.6E+00	1
(X2)	GMW	20	299	58	1.4E+01	6.6E+01	4.0E+01	2
	S&E,(a)	18	278	45	7.2E+00	7.2E+00	7.2E+00	1
	S&E,(b)	18	287	36	7.2E+00	7.2E+00	7.2E+00	1
(X3)	GMW	13	189	34	1.3E+01	1.3E+01	1.3E+01	1
	S&E,(a)	12	156	23	4.7E+00	7.2E+00	6.0E+00	2
	S&E,(b)	13	177	31	7.2E+00	7.2E+00	7.2E+00	1

Table III

Eigenvalue Information for dC Starting Points

Point		λmin	λmax	#NEG
(X1)	H	-1.4E+00	2.9E+03	2
	M	-3.2E-01	2.8E+03	1
(X2)	H	-1.2E+04	1.8E+05	7
	M	-1.2E-01	2.9E+03	1
(X3)	H	-8.2E+05	1.1E+07	9
	M	-1.2E-01	2.8E+03	1

F(X1) = 1.4E+00
F(X2) = 1.8E+03
F(X3) = 5.1E+04

Table IV

Eigenvalue Information for dC Minima

Point		λmin	λmax	#NEG
(X5)	H	7.7E-02	3.0E+03	0
	M	3.7E-01	3.0E+03	0
(X6)	H	3.8E-02	2.9E+03	0
	M	2.6E-01	2.9E+03	0
(X7)	H	2.1E-02	3.1E+03	0
	M	3.8E-01	3.1E+03	0

F(X5) = -6.8E+00
F(X6) = -5.7E+00
F(X7) = -5.0E+00

Table V

Molecular Model 2: Water Dimer, n = 18

	Run	Newton Itns.	PCG Itns.	F&G Evals.	$\|\|E\|\|_\infty$ Min.	$\|\|E\|\|_\infty$ Max.	$\|\|E\|\|_\infty$ Med.	Nonzero Itns.
	GMW	38	244	100	1.5E-05	1.6E-05	1.6E-05	38
(A)	S&E,(a)	47	322	132	9.5E-01	1.1E+03	5.5E+02	47
	S&E,(b)	43	296	120	9.5E-01	1.1E+03	5.5E+02	43
	GMW	46	294	129	1.0E+01	6.7E+03	3.4E+03	46
(B)	S&E,(a)	45	297	127	6.0E+00	9.2E+02	4.6E+02	45
	S&E,(b)	45	297	134	1.5E+01	9.2E+02	4.7E+02	45

Table VI-(A)

MCF Analysis for the First Newton Iteration of Table V, Run A

j	GMW d_j	GMW d_j+e_j	S&E,(a) d_j	S&E,(a) d_j+e_j	S&E,(b) d_j	S&E,(b) d_j+e_j
10			350	1452	350	1452
11			1102	2105	1102	2005
12	-2.1E-13	1.5E-05	266	1368	266	837
13	3.6E-15	1.5E-05	86	1189	77	230
14	1.6E+02	0	349	1451	288	288
15	-5.0E-14	1.5E-05	71	1173	31	62
16	3.6E-15	1.5E-05	86	1188	72	201
17	-3.6E-15	1.5E-05	341	1444	209	209
18	-5.0E-14	1.5E-05	70	1172	13	13
	$\|\|E\|\|_\infty$ =1.5E-05		$\|\|E\|\|_\infty$ =1103		$\|\|E\|\|_\infty$ =1103	

Table VI-(B)

MCF Analysis for the Tenth Newton Iteration of Table V, Run B

j	GMW d_j	GMW d_j+e_j	S&E,(a) d_j	S&E,(a) d_j+e_j	S&E,(b) d_j	S&E,(b) d_j+e_j
14	*		83	128	94	169
15	- 0.2	0.2	0.3	46	3	20
16	*		12	58	12	15
17	- 13	13	22	68	*	
18	-104	104	- 0.5	45	- 0.6	0.02
	$\|\|E\|\|_\infty$ = 208		$\|\|E\|\|_\infty$ = 46		$\|\|E\|\|_\infty$ = 75	

* Entries leading to zero modifications were not recorded

- T. Schlick -

Table VII

Molecular Model 2: Water Clusters

n	Run	Newton Itns.	PCG Itns.	F&G Evals.	Min.	$\|\|E\|\|_\infty$ Max.	Med.	Nonzero Itns.
	GMW	38	506	67	1.5E-05	6.6E-01	3.3E-01	38
(27)	S&E,(a)	63	1147	94	3.3E+02	1.6E+03	1.0E+03	63
	S&E,(b)	68	1053	106	5.0E+02	1.5E+03	1.0E+03	68
	GMW	70	1400	114	1.5E-05	1.8E-05	1.7E-05	70
(36)	S&E,(a)	71	1783	110	3.9E+02	1.5E+03	9.5E+02	71
	S&E,(b)	75	1554	128	3.9E+02	1.6E+03	1.0E+03	75
	GMW	81	2256	148	1.5E-05	1.9E-05	1.7E-05	81
(45)	S&E,(a)	99	3606	159	5.4E+02	1.6E+03	1.1E+03	99
	S&E,(b)	94	2598	162	9.6E+02	1.5E+03	1.2E+03	94
	GMW	116	4744	212	1.5E-05	1.9E-05	1.7E-05	116
(54)	S&E,(a)	116	5554	179	9.9E+02	1.6E+03	1.3E+03	116
	S&E,(b)	108	3886	158	1.1E+03	1.6E+03	1.4E+03	108
	GMW	85	4771	157	1.5E-05	1.9E-05	1.7E-05	85
(72)	S&E,(a)	64	3155	102	8.0E+02	1.6E+03	1.2E+03	64
	S&E,(b)	54	1848	89	8.4E+02	1.6E+03	1.2E+03	54
	GMW	112	8524	200	1.5E-05	1.9E-05	1.7E-05	112
(90)	S&E,(a)	85	5804	133	1.0E+03	1.6E+03	1.3E+03	85
	S&E,(b)	85	4879	154	1.1E+03	1.6E+03	1.3E+03	85
	GMW	143	12941	270	1.5E-05	1.3E+01	6.4E+00	143
(108)	S&E,(a)	101	7573	162	1.2E+03	1.5E+03	1.4E+03	101
	S&E,(b)	97	6285	174	1.0E+03	1.7E+03	1.4E+03	97
	GMW	132	15702	249	1.6E-05	2.3E-05	1.9E-05	132
(144)	S&E,(a)	118	12217	217	1.2E+03	1.6E+03	1.4E+03	118
	S&E,(b)	97	8631	163	1.2E+03	1.6E+03	1.4E+03	97
	GMW	88	10314	236	1.5E-05	1.8E-05	1.6E-05	88
(180)	S&E,(a)	47	3816	104	1.3E+03	1.6E+03	1.5E+03	47
	S&E,(b)	47	2657	107	1.1E+03	1.7E+03	1.4E+03	47
	GMW	71	8173	212	1.4E-05	1.7E-05	1.6E-05	71
(243)	S&E,(a)	56	5828	104	1.1E+03	1.6E+03	1.4E+03	56
	S&E,(b)	60	4880	137	1.1E+03	1.8E+03	1.5E+03	60
	GMW	104	20543	277	1.4E-05	1.9E-05	1.7E-05	104
(324)	S&E,(a)	53	4450	120	1.2E+03	1.7E+03	1.5E+03	53
	S&E,(b)	55	4581	123	1.1E+03	1.7E+03	1.4E+03	55
	GMW	100	23502	407	1.4E-05	1.6E-05	1.5E-05	100
(576)	S&E,(a)	66	9040	195	1.3E+03	1.6E+03	1.5E+03	66
	S&E,(b)	57	7732	139	1.4E+03	1.7E+03	1.6E+03	57

Table VIII

Eigenvalue Information for Water Clusters

n		X0 λmin	X0 λmax	X0 #NEG	X* λmin	X* λmax	X* #NEG
(18)	H	-2.4E+00	2.2E+03	3	9.8E-02	2.2E+03	0
	M	-1.6E-13	2.1E+03	1	-1.7E-14	2.2E+03	2
(27)	H	-4.5E+01	2.3E+03	8	5.4E-02	2.3E+03	0
	M	-1.3E-13	2.3E+03	2	-1.2E-14	2.2E+03	2
(36)	H	-4.5E+01	2.3E+03	11	1.5E-02	2.3E+03	0
	M	-1.5E-13	2.3E+03	3	-2.8E-14	2.2E+03	5
(45)	H	-4.5E+01	2.3E+03	14	1.9E-04	2.3E+03	0
	M	-1.5E-13	2.3E+03	4	-2.8E-14	2.2E+03	4
(54)	H	-5.6E+01	2.3E+03	17	9.6E-03	2.3E+03	0
	M	-1.5E-13	2.3E+03	7	-2.1E-14	2.2E+03	5
(72)	H	-5.6E+01	2.3E+03	23	8.9E-03	2.3E+03	0
	M	-1.5E-13	2.3E+03	9	-3.6E-14	2.2E+03	6
(90)	H	-1.1E+02	2.3E+03	28	1.2E-03	2.3E+03	0
	M	-1.5E-13	2.3E+03	12	-2.8E-14	2.2E+03	7
(108)	H	-7.5E+01	2.3E+03	34	4.1E-03	2.3E+03	0
	M	-1.6E-13	2.3E+03	12	-2.8E-14	2.2E+03	9
(144)	H	-9.6E+01	2.3E+03	45	8.3E-03	2.3E+03	0
	M	-1.5E-13	2.3E+03	17	-3.4E-14	2.2E+03	14
(180)	H	-1.1E+01	2.4E+03	54	3.3E-03	2.3E+03	0
	M	-1.7E-13	2.3E+03	24	-3.3E-14	2.2E+03	16
(243)	H	-1.1E+02	2.4E+03	70	2.6E-03	2.3E+03	0
	M	-2.5E-13	2.3E+03	31	-2.9E-14	2.2E+03	25

Table IX

Trigonometric Function Minimization

n = 50

M	Run	Newton Itns.	PCG Itns.	F&G Evals.	$\|\|E\|\|_\infty$ Min.	$\|\|E\|\|_\infty$ Max.	$\|\|E\|\|_\infty$ Med.	Nonzero Itns.
	GMW	15	79	22	1.1E+00	1.1E+00	1.1E+00	1
(D)	S&E,(a)	15	88	41	5.5E-01	5.5E-01	5.5E-01	1
	S&E,(b)	22	84	38	2.1E-03	5.5E-01	2.8E-01	9
	GMW	22	102	54	3.0E-02	1.5E+00	7.4E-01	14
(T)	S&E,(a)	15	108	20	4.2E-02	6.0E+00	3.2E-01	7
	S&E,(b)	13	107	18	3.7E-02	6.0E-01	3.2E-01	6
	GMW	30	125	52	8.0E-02	6.1E+00	3.1E+00	22
(P)	S&E,(a)	14	111	38	9.0E-02	6.2E-01	3.5E-01	6
	S&E,(b)	13	101	17	5.7E-02	6.2E-01	3.4E-01	8

n = 100

M	Run	Newton Itns.	PCG Itns.	F&G Evals.	$\|\|E\|\|_\infty$ Min.	$\|\|E\|\|_\infty$ Max.	$\|\|E\|\|_\infty$ Med.	Nonzero Itns.
	GMW	19	65	36	1.3E-01	1.1E+00	6.3E-01	13
(D)	S&E,(a)	47	109	83	3.0E-03	5.7E-01	2.8E-01	34
	S&E,(b)	30	79	41	6.9E-03	5.7E-01	2.9E-01	15
	GMW	28	114	54	6.3E-02	1.7E+00	8.7E-01	19
(T)	S&E,(a)	15	98	22	2.0E-02	5.9E-01	3.1E-01	8
	S&E,(b)	15	107	36	2.8E-02	5.9E-01	3.1E-01	7
	GMW	58	139	92	8.8E-02	7.7E+00	3.9E+00	51
(P)	S&E,(a)	14	100	93	4.4E-02	6.0E-01	3.2E-01	7
	S&E,(b)	13	100	21	2.4E-02	6.0E-01	3.1E-01	6

n = 250

M	Run	Newton Itns.	PCG Itns.	F&G Evals.	$\|\|E\|\|_\infty$ Min.	$\|\|E\|\|_\infty$ Max.	$\|\|E\|\|_\infty$ Med.	Nonzero Itns.
	GMW	35	106	91	1.1E-01	1.2E+00	5.8E-01	28
(D)	S&E,(a)	47	134	71	4.3E-02	5.8E-01	2.9E-01	33
	S&E,(b)	126	197	183	3.2E-05	5.8E-01	2.9E-01	115
	GMW	45	138	88	8.4E-03	1.4E+00	7.2E-01	36
(T)	S&E,(a)	20	109	54	7.1E-02	5.9E-01	3.0E-01	10
	S&E,(b)	61	136	100	6.1E-03	5.9E-01	3.0E-01	52
	GMW	50	166	106	1.5E-02	7.8E+00	3.9E+00	40
(P)	S&E,(a)	17	108	40	2.0E-02	5.9E-01	3.1E-01	8
	S&E,(b)	23	124	56	2.0E-02	5.9E-01	3.1E-01	11

-- Table IX, cont. --

n =1000

	GMW	55	151	128	2.2E-03	1.7E+00	8.3E-01	49
(D)	S&E,(a)	150	220	229	3.1E-05	5.8E-01	2.9E-01	137
	S&E,(b)	223	276	319	2.9E-03	5.8E-01	2.9E-01	215
	GMW	86	206	181	2.6E-01	1.8E+00	1.0E+00	79
(T)	S&E,(a)	34	95	74	4.5E-03	5.8E-01	2.9E-01	26
	S&E,(b)	145	184	205	4.5E-03	5.8E-01	2.9E-01	140
	GMW	50	133	126	6.7E-02	1.9E+00	9.9E-01	46
(P)	S&E,(a)	21	94	60	5.9E-03	5.9E-01	3.0E-01	12
	S&E,(b)	41	109	61	5.9E-03	5.9E-01	3.0E-01	33

The symbols (D), (T), and (P) denote Diagonal, Tridiagonal, and Pentadiagonal Preconditioners, respectively

Table X

Eigenvalue Information for the Trigonometric Function

n		X0 λmin	X0 λmax	X0 #NEG	X* λmin	X* λmax	X* #NEG
(50)	H	-5.7E-01	1.4E+00	30	5.3E-02	2.7E+00	0
	M	-6.0E-01	1.4E+00	31	6.4E-02	2.2E+00	0
(100)	H	-5.8E-01	1.5E+00	60	6.3E-02	2.8E+00	0
	M	-5.9E-01	1.4E+00	61	6.5E-02	2.2E+00	0
(250)	H	-5.8E-01	1.5E+00	152	4.5E-02	2.8E+00	0
	M	-5.9E-01	1.5E+00	152	6.3E-02	2.0E+00	0
(1000)	H	-5.8E-01	1.5E+00	607	4.8E-02	2.7E+00	0
	M	-5.8E-01	1.5E+00	608	7.6E-02	2.0E+00	0

NYU COMPSCI TR-525
Schlick, Tamar
Implementation of the
Schnabel & Eskow modified
Cholesky factorization in

DATE DUE	BORROWER'S NAME

DATE DUE